Eurographics 2009

Advanced Illumination Techniques for GPU-Based Volume Raycasting

Markus Hadwiger
VRVis Research Center
Vienna, Austria

Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer Graphics Research Group,
University of Münster, Germany

Eurographics 2009

Ray Casting Basics

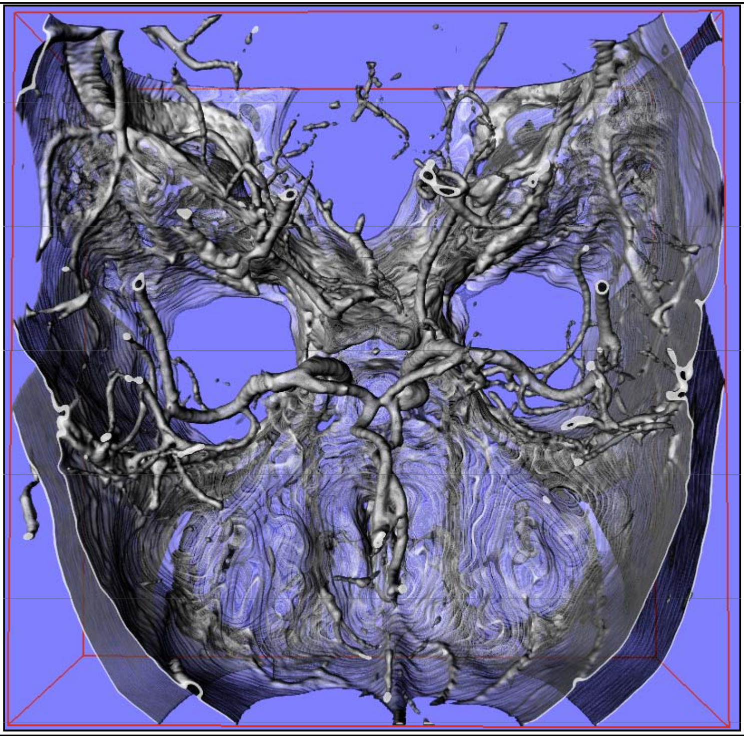
Markus Hadwiger
VRVis Research Center
Vienna, Austria

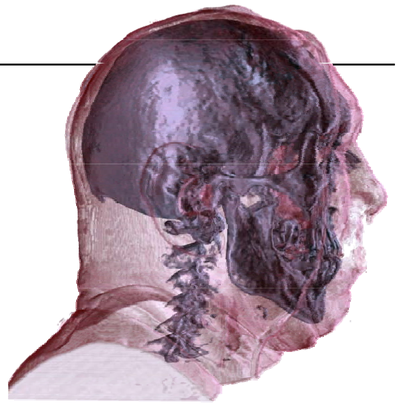
Patric Ljung
Siemens Corporate Research
Princeton, NJ, USA

Christof Rezk Salama
Computer Graphics Group
Institute for Vision and Graphics
University of Siegen, Germany

Timo Ropinski
Visualization and Computer Graphics Research Group,
University of Münster, Germany

Medicine

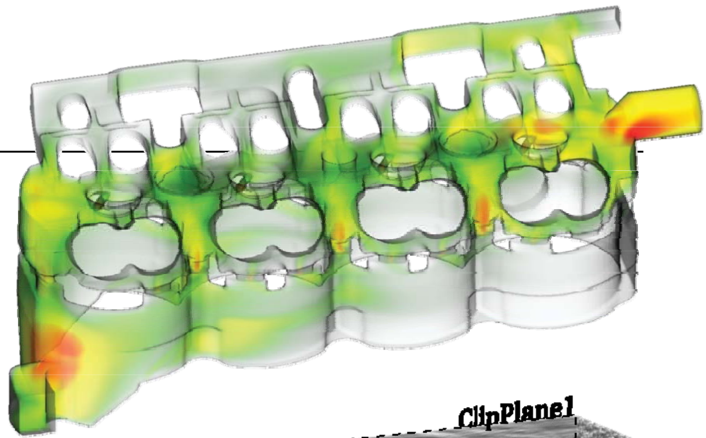




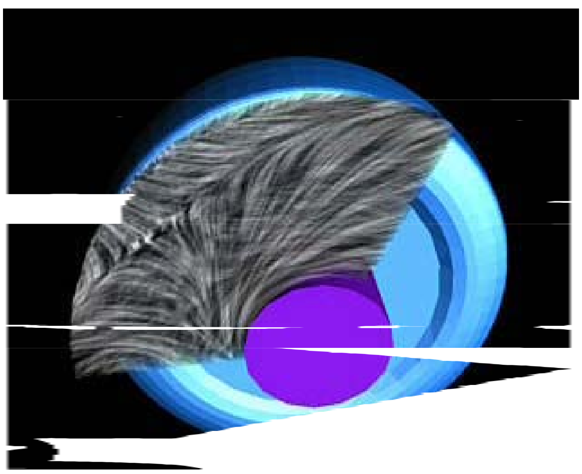
CT Human Head:
Visible Human Project,
US National Library of Medicine,
Maryland, USA

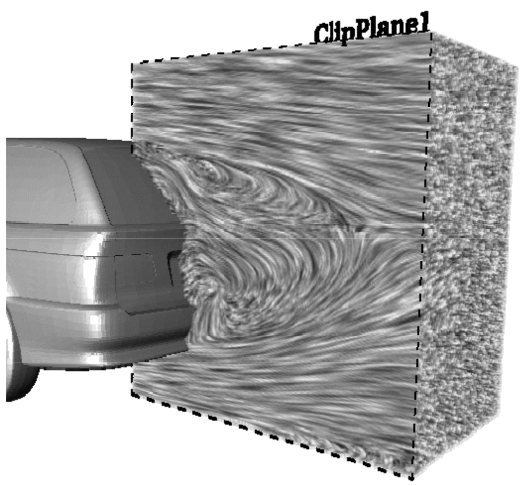
CT Angiography:
Dept. of Neuroradiology
University of Erlangen,
Germany

Engineering



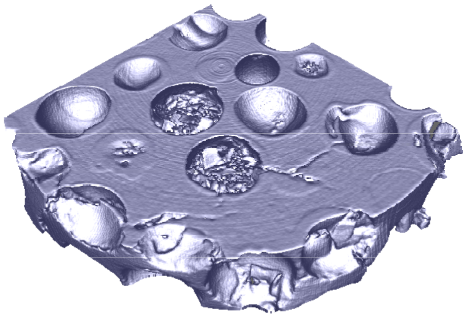
Computational Fluid Dynamics (CFD)



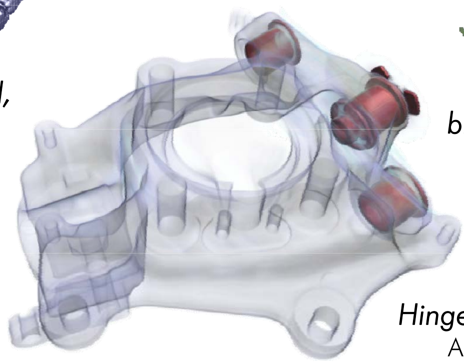


Materials Science, Biology

Materials Science, NDT

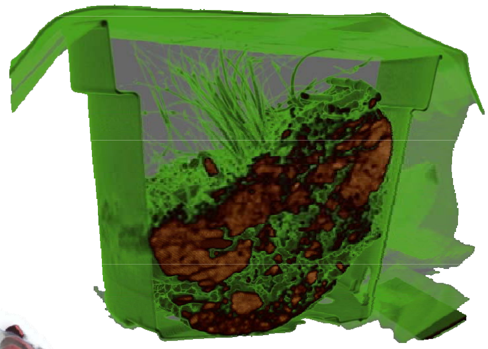


Micro CT, Compound Material,
Material Science Department,
University of Erlangen



Hinge Bearing,
Austrian Foundry Research Institute

Biology



biological sample of the soil, CT,
Virtual Reality Group,
University of Erlangen

Archaeology

Hellenic Statue of Isis
3rd century B.C.
ARTIS, University of Erlangen-Nuremberg, Germany

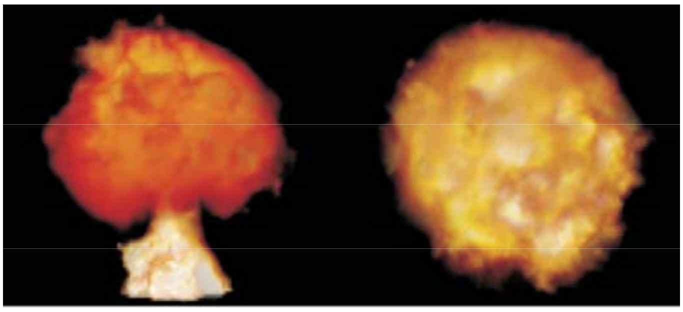
Sotades Pygmaios Statue,
5th century B.C
ARTIS, University of Erlangen-Nuremberg, Germany

Special Effects and Games

Clouds and Atmospheric Scattering

Dobashi et al.

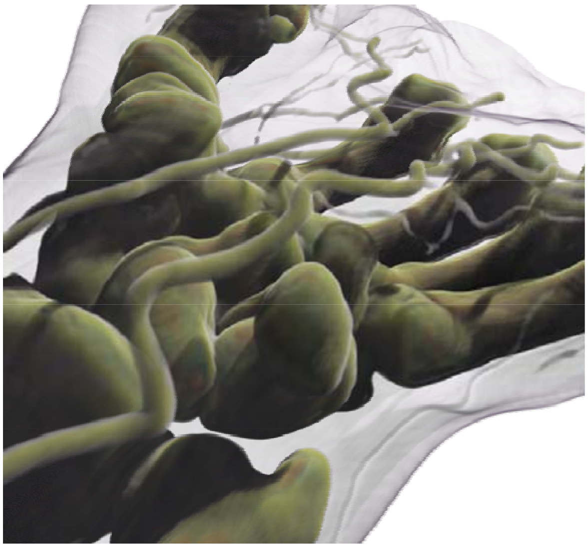
Fire and Explosions



Krüger and Westermann

Advanced Lighting

- **Shadows and scattering**



Advanced Lighting

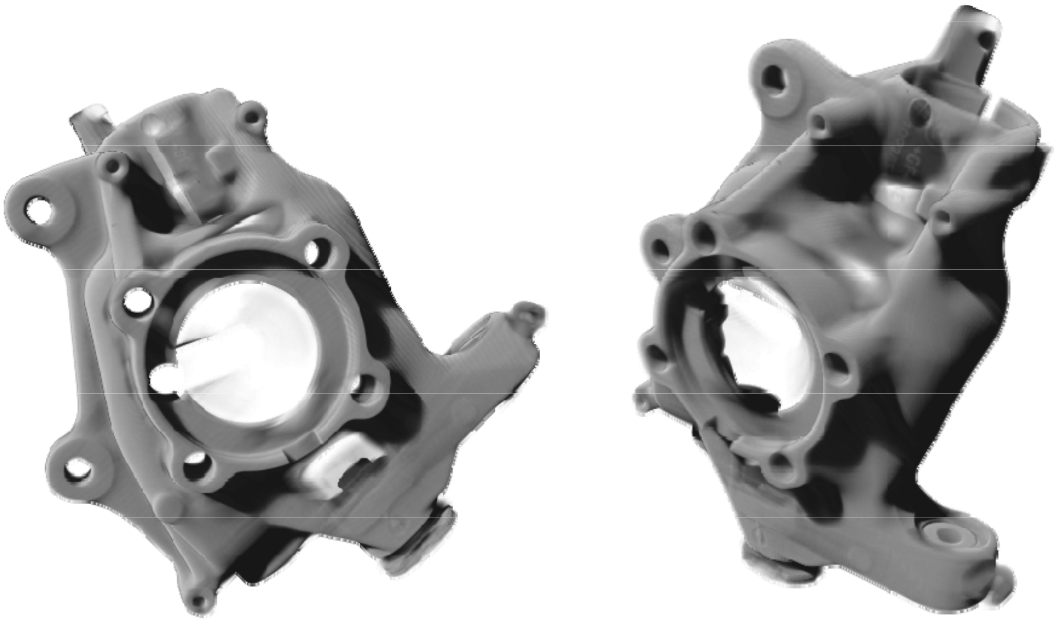
- **MRI Brain**

no shading | gradient shading | shadows+scattering

Advanced Lighting

Advanced Lighting

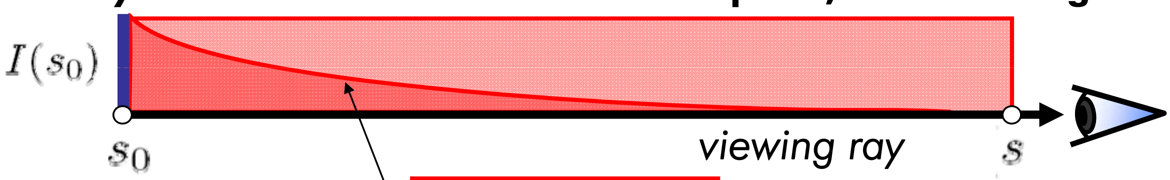
- **Industrial CT**



Ray Integration

How do we determine the radiant energy along the ray?

Physical model: **emission and absorption, no scattering**



Absorption along the ray segment s_0 - s

Initial intensity at s_0

$$I(s) = I(s_0)\, e^{-\tau(s_0,s)}$$

Extinction τ

Absorption κ

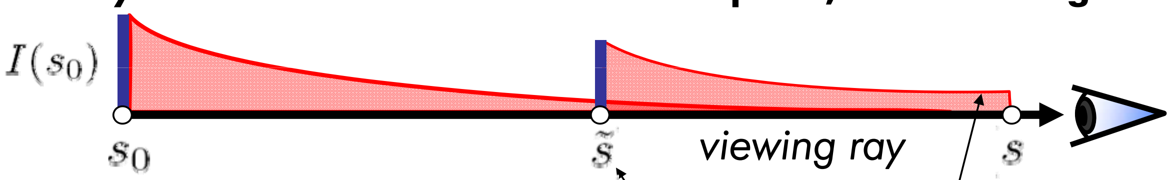
Without absorption all the initial radiant energy would reach the point s.

$$\tau(s_1, s_2) = \int_{s_1}^{s_2} \kappa(s)\, ds.$$

Ray Integration

How do we determine the radiant energy along the ray?

Physical model: **emission and absorption, no scattering**



Every point $\tilde{s}$ along the viewing ray emits additional radiant energy.

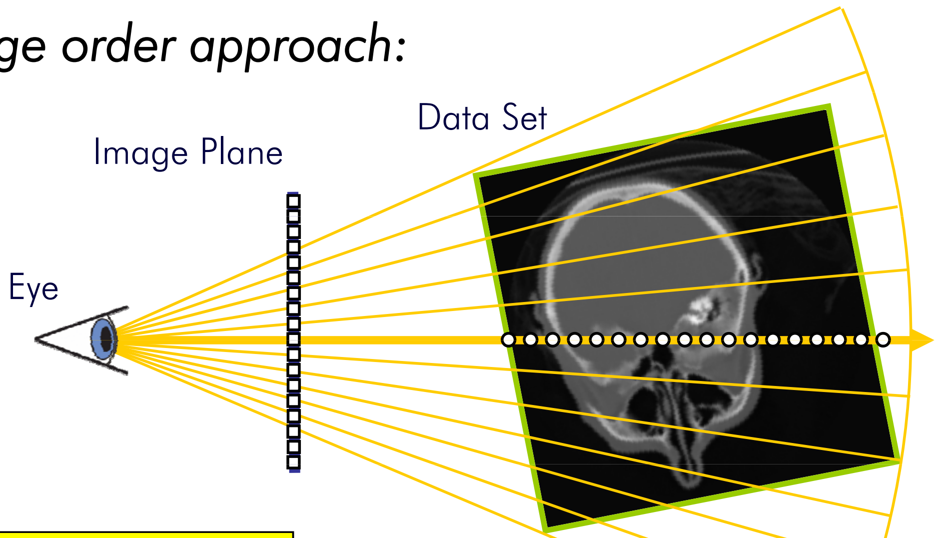
Active emission at point $\tilde{s}$

Absorption along the distance $s - \tilde{s}$

$$I(s) = I(s_0)\, e^{-\tau(s_0,s)} + \int_{s_0}^{s} q(\tilde{s})\, e^{-\tau(\tilde{s},s)}\, d\tilde{s}$$

Volume Rendering

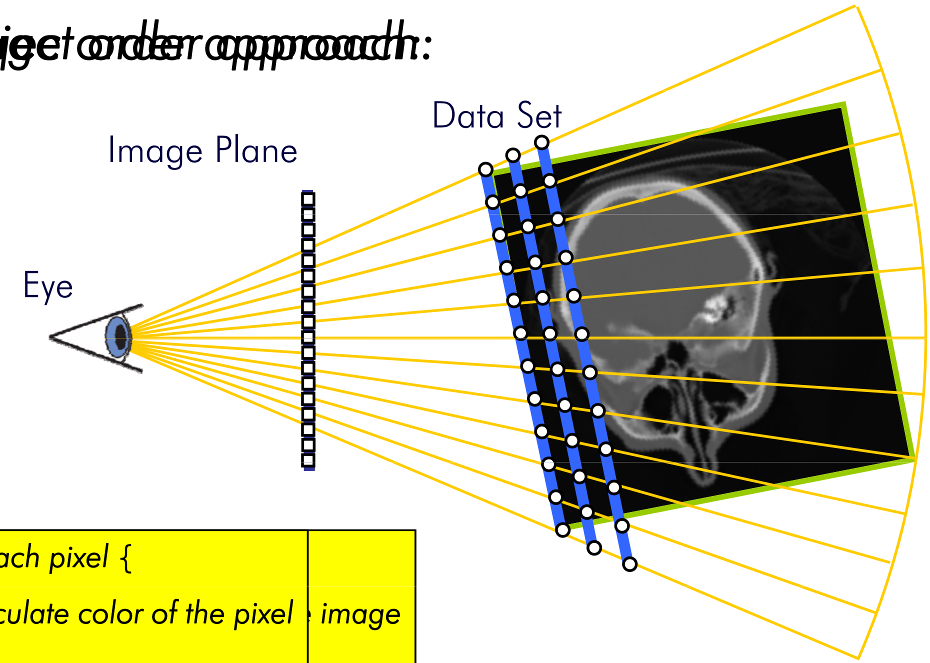
Image order approach:




```
For each pixel {
  calculate color of the pixel
}
```

Volume Rendering

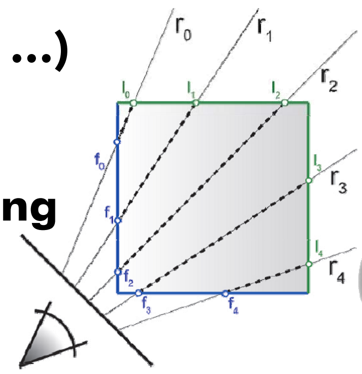
Image order approach:
Object order approach:

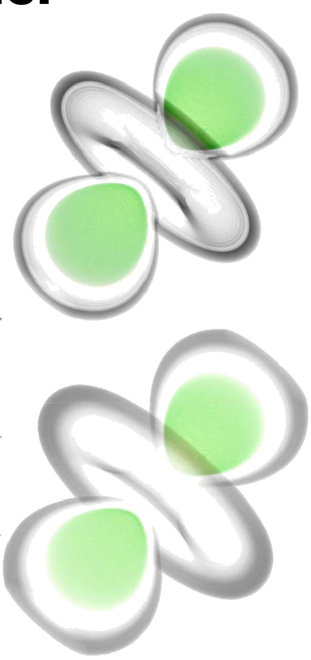


For each pixel {
calculate color of the pixel
}
image

Why Ray Casting on GPUs?

- **Most GPU rendering is object-order (rasterization)**
- **Image-order is more "CPU-like"**
  - **Simpler to implement**
  - **Very flexible (adaptive sampling, ...)**
  - **Correct perspective**
  - **Single pass ray casting**
  - **32-bit compositing**

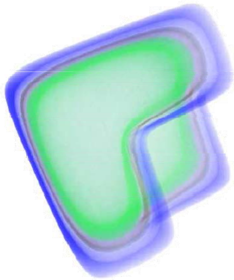




Recent GPU Approaches

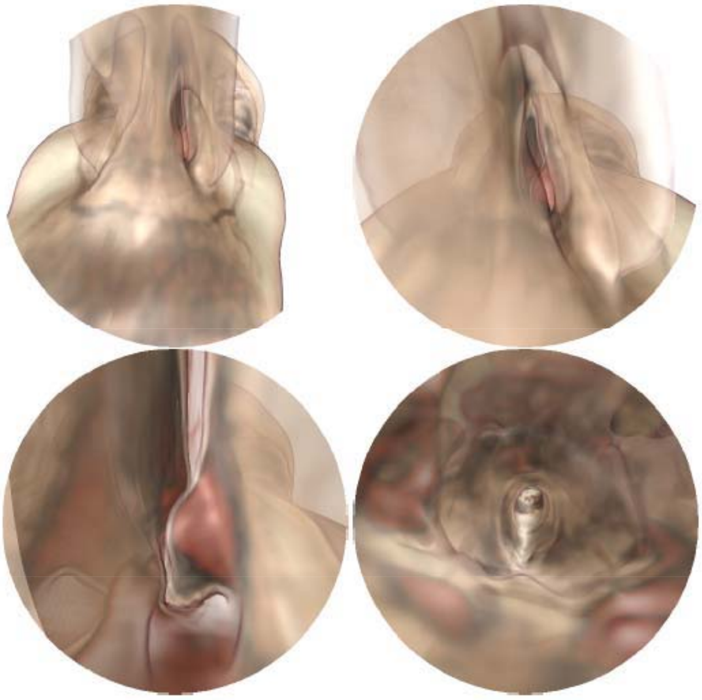
- **Rectilinear grids**
  - [Krüger and Westermann, 2003]
  - [Röttger et al., 2003]
  - [Green, 2004] (in NVIDIA SDK)
  - [Stegmaier et al., 2005]
  - [Scharsach et al., 2006]
  - [Gobbetti et al., 2008]

- **Unstructured (tetrahedral) grids**
  - [Weiler et al., 2002, 2003, 2004]
  - [Bernardon et al., 2004]
  - [Callahan et al., 2006]
  - [Muigg et al., 2007]



Correct Perspective

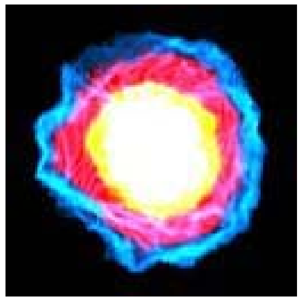
- **Entering the volume**
- **Wide field of view**
- **Fly-throughs**
- **Virtual endoscopy**
- **Integration into perspective scenes: games, …**



Single-Pass Ray Casting

- **Enabled by conditional loops in fragment shaders (Shader Model 3.0 and higher)**
- **Substitute multiple passes and early-z testing by single loop and early loop exit**
- **No compositing buffer: full 32-bit precision!**

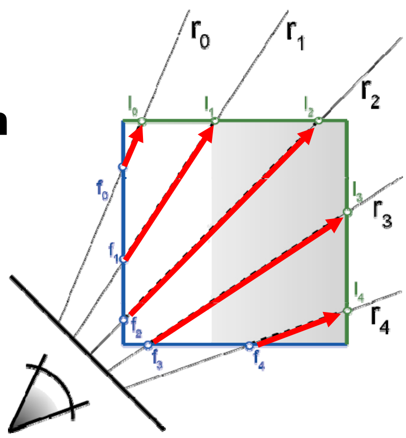
- **NVIDIA SDK example: compute ray intersections with bounding box, march along rays and composite**



- **Volume rendering example in NVIDIA CUDA SDK**

Basic Ray Setup / Termination

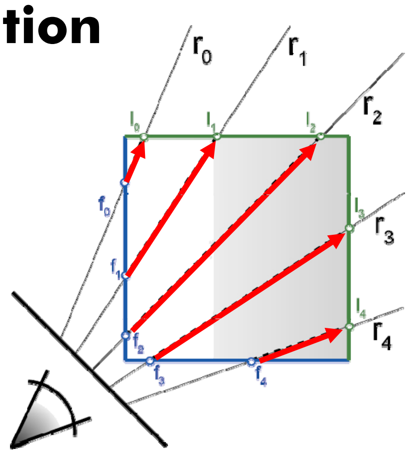
- **Two main approaches:**
  - **Procedural ray/box intersection**
    [Röttger et al., 2003], [Green, 2004]
  - **Rasterize bounding box**
    [Krüger and Westermann, 2003]



- **Either:**
  - **Ray start position and exit check**
  - **Ray start position and exit position**
  - **Ray start position and direction vector**

Procedural Ray Setup / Term.

- **Procedural ray / box intersection**
  - **Everything handled in fragment shader**



- **Ray given by camera position and volume entry position**
- **Exit criterion needed**

- **Pro: simple and self-contained**
- **Con: full load on fragment shader**

Fragment Shader

- **Rasterize front faces of bounding box**

- **Texcoords are volume position in [0,1]**
- **Subtract camera pos**

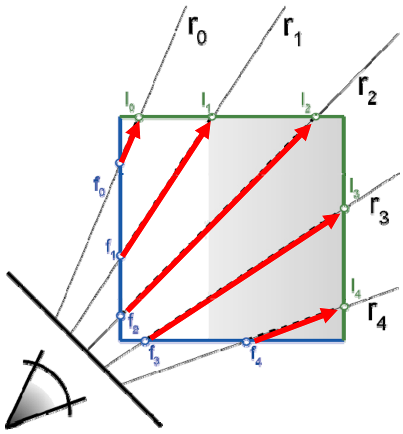
- **Accumulate/composite**

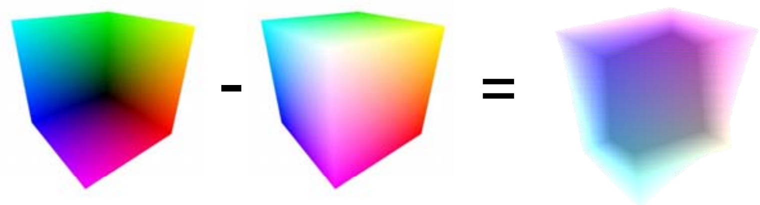
- **Repeatedly check for exit of bounding box**

```
// Cg fragment shader code for single-pass ray casting
float4 main(VS_OUTPUT IN, float4 TexCoord0 : TEXCOORD0,
            uniform sampler3D SamplerDataVolume,
            uniform sampler1D SamplerTransferFunction,
            uniform float3 camera,
            uniform float stepsize,
            uniform float3 volExtentMin,
            uniform float3 volExtentMax
            ) : COLOR
{
    float4 value;
    float scalar;
    // Initialize accumulated color and opacity
    float4 dst = float4(0,0,0,0);
    // Determine volume entry position
    float3 position = TexCoord0.xyz;
    // Compute ray direction
    float3 direction = TexCoord0.xyz - camera;
    direction = normalize(direction);
    // Loop for ray traversal
    for (int i = 0; i < 200; i++)  // Some large number
    {
        // Data access to scalar value in 3D volume texture
        value = tex3D(SamplerDataVolume, position);
        scalar = value.a;
        // Apply transfer function
        float4 src = tex1D(SamplerTransferFunction, scalar);
        // Front-to-back compositing
        dst = (1.0-dst.a) * src + dst;
        // Advance ray position along ray direction
        position = position + direction * stepsize;
        // Ray termination:  Test if outside volume ...
        float3 temp1 = sign(position - volExtentMin);
        float3 temp2 = sign(volExtentMax - position);
        float inside = dot(temp1, temp2);
        // ... and exit loop
        if (inside < 3.0)
            break;
    }
    return dst;
}
```

"Image-Based" Ray Setup / Term.

- Rasterize bounding box
  front faces and back faces
- Ray start positions:
  front faces
- Direction vectors:
  back faces – front faces



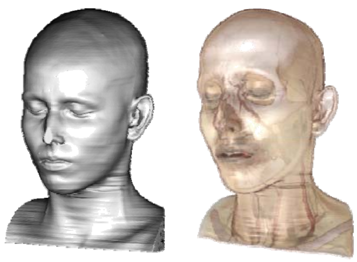


- Independent of projection (orthogonal/perspective)

Standard Ray Casting Optim. (1)

Early ray termination

- Isosurfaces:
  stop when surface hit
- Direct volume rendering:
  stop when opacity >= threshold

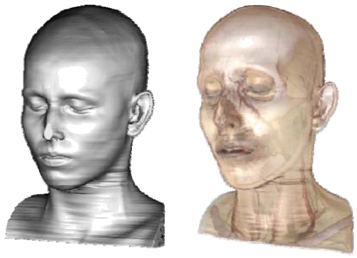


- Several possibilities
  - Older GPUs (before shader model 3):
    multi-pass rendering with early-z test
  - Shader model 3: break out of ray-casting loop
  - Current GPUs: early loop exit works well

Standard Ray Casting Optim. (2)

Empty space skipping

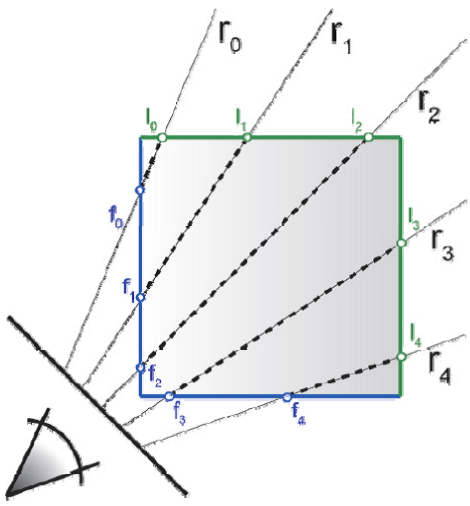
- Skip transparent samples
- Depends on transfer function
- Start casting close to first hit



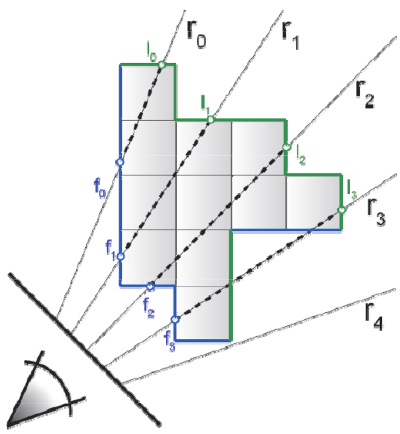
- Several possibilities
  - Per-sample check of opacity (expensive)
  - Traverse regular grid or hierarchy (e.g., octree with stack-less traversal [Gobbetti et al., 2008])
  - These are image-order: what about object-order?

Object-Order Empty Space Skip. (1)

- Modify initial rasterization step



rasterize bounding box



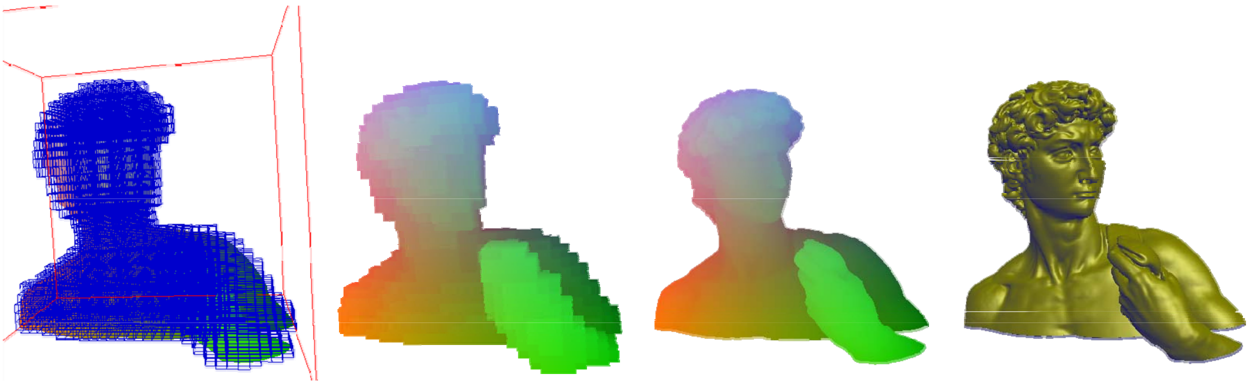
rasterize “tight" bounding geometry

ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING

Eurographics 2009

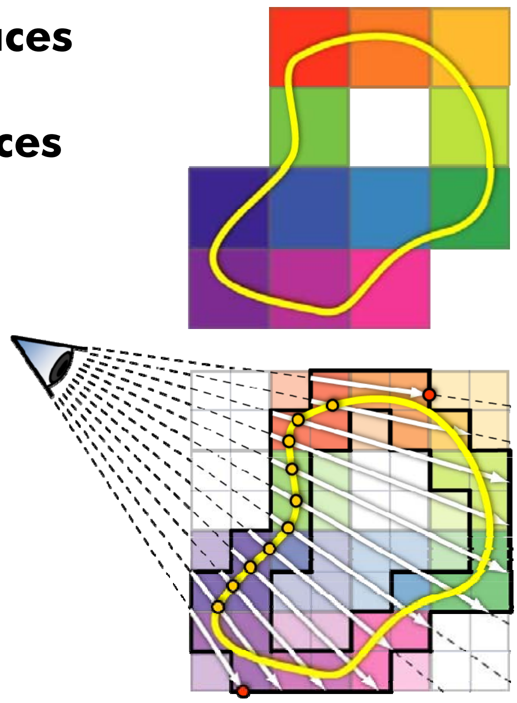
Object-Order Empty Space Skip. (2)

- Store min-max values of volume blocks
- Cull blocks against transfer function or isovalue
- Rasterize front and back faces of active blocks



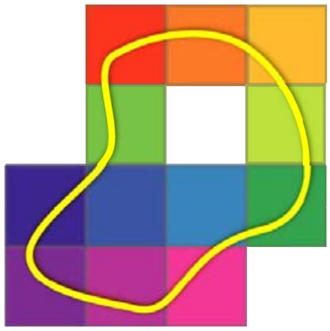
Object-Order Empty Space Skip. (3)

- Rasterize front and back faces of active min-max blocks
- Start rays on block front faces
- Terminate when
  - Full opacity reached, or
  - Back face reached

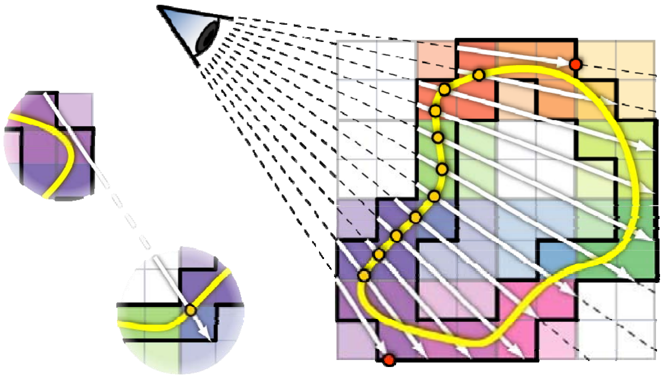


Object-Order Empty Space Skip. (3)

- Rasterize front and back faces of active min-max blocks
- Start rays on block front faces
- Terminate when
  - Full opacity reached, or
  - Back face reached



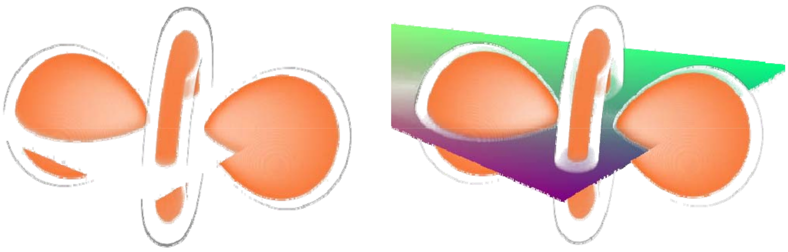
- Not all empty space is skipped



Scene Integration (1)

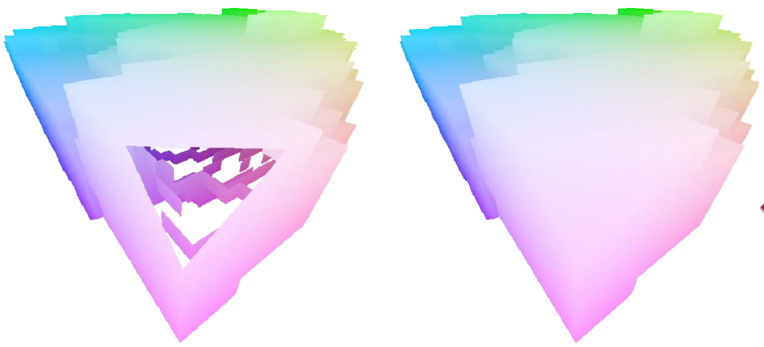
- Build on image-based ray setup
- Allow viewpoint inside the volume

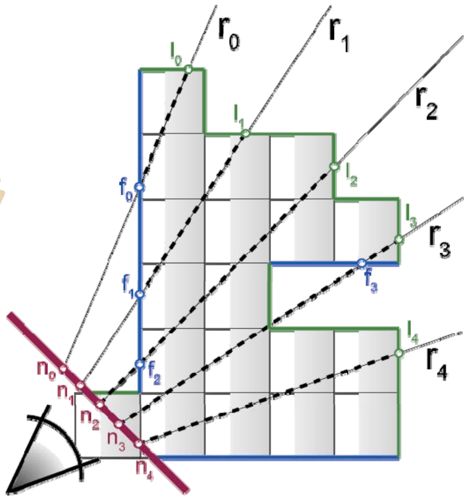
- Intersect polygonal geometry



Scene Integration (2)

- Near clipping plane clips into front faces

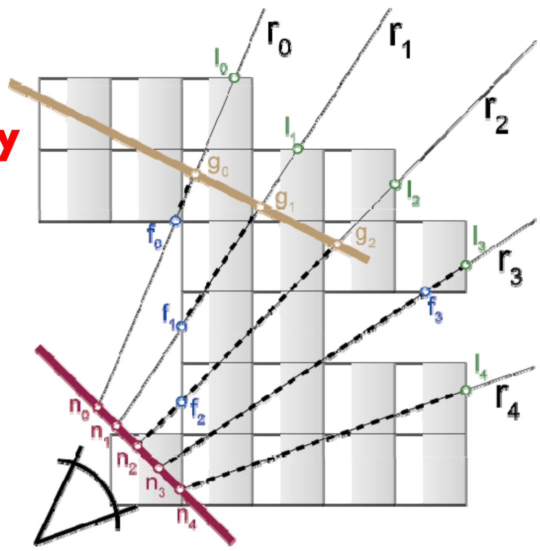




- Fill in holes with near clipping plane
- Can use depth buffer [Scharsach et al., 2006]

Scene Integration (3)

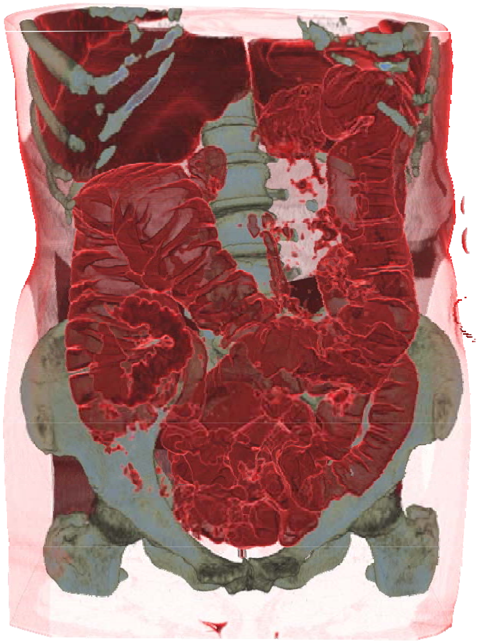
1. **Starting position computation**
   ⇨ Ray start position image
2. Ray length computation
   ⇨ Ray length image
3. **Render polygonal geometry**
   ⇨ Modified ray length image
4. Raycasting
   ⇨ Compositing buffer
5. Blending
   ⇨ Final image



Virtual Endoscopy

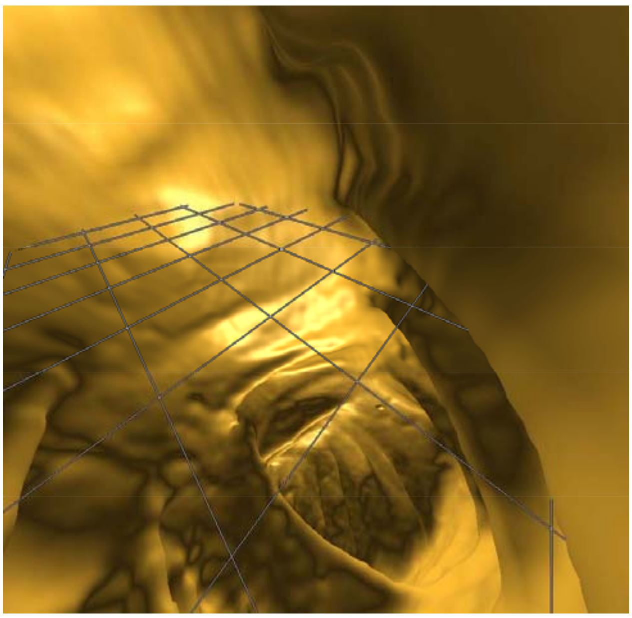
- Viewpoint inside the volume with wide field of view
- E.g.: virtual colonoscopy

- Hybrid isosurface rendering / direct volume rendering
- E.g.: colon wall and structures behind



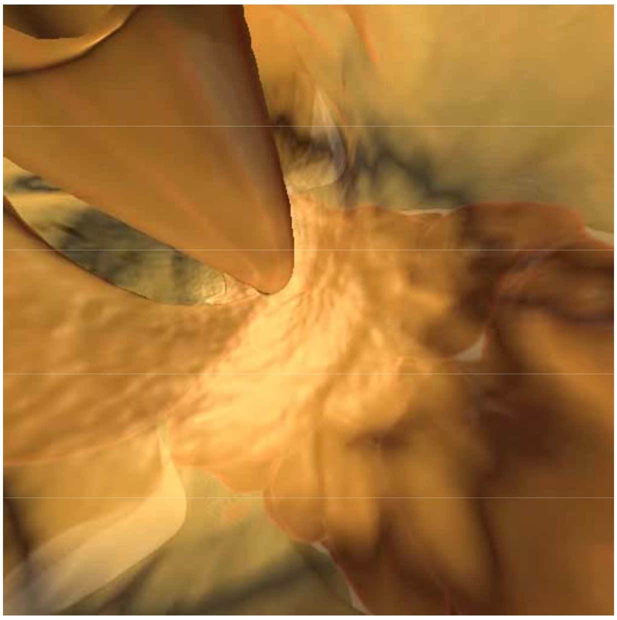
Virtual Colonoscopy

- First find isosurface; then continue with DVR



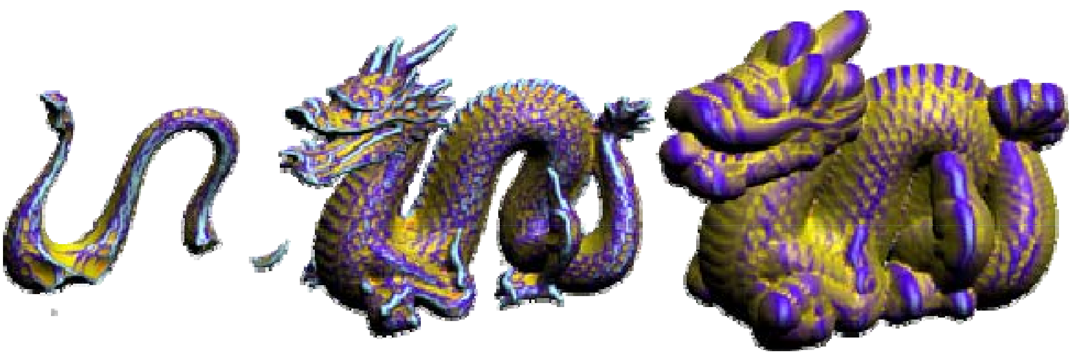
Virtual Colonoscopy

- First find isosurface; then continue with DVR



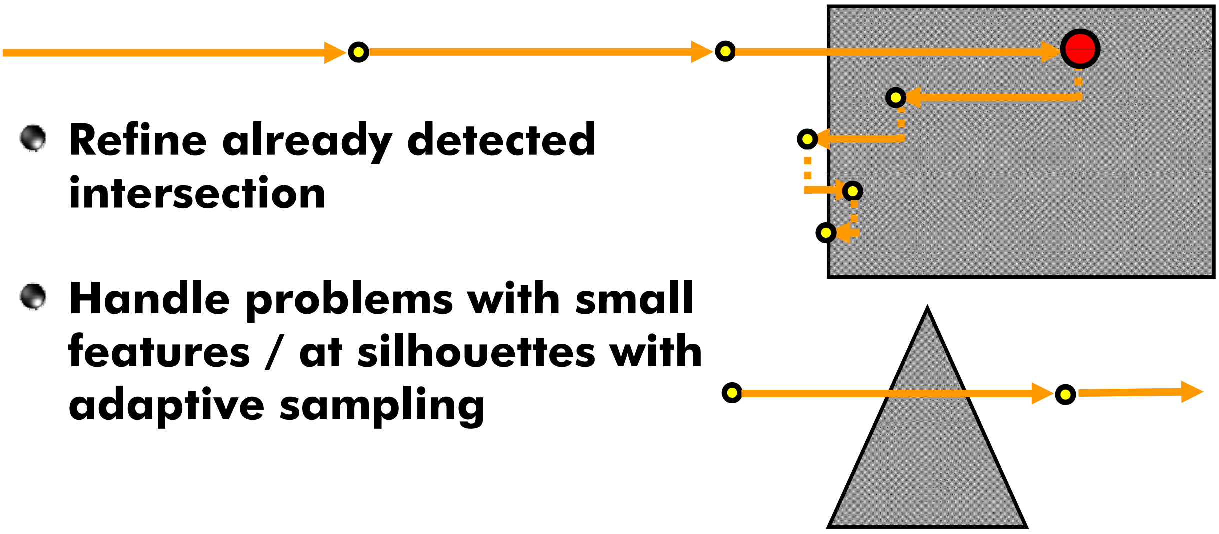
Isosurface Ray Casting

- Isosurfaces/Level Sets
  - scanned data
  - distance fields
  - CSG operations
  - level sets: surface editing, simulation, segmentation, …



Intersection Refinement (1)

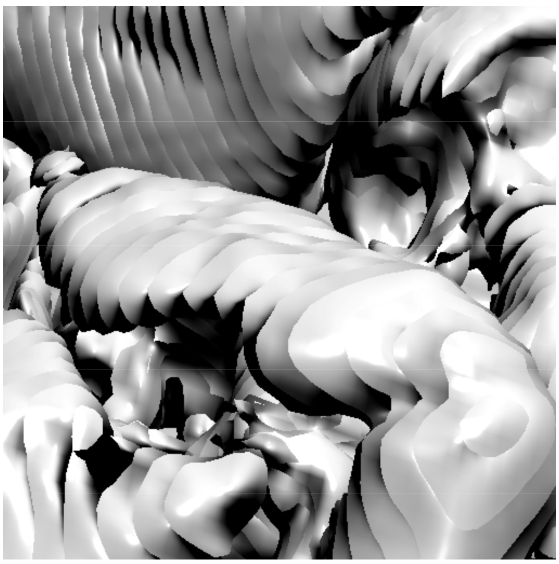
- Fixed number of bisection or binary search steps
- Virtually no impact on performance



- Refine already detected intersection
- Handle problems with small features / at silhouettes with adaptive sampling

Intersection Refinement (2)

without refinement



with refinement

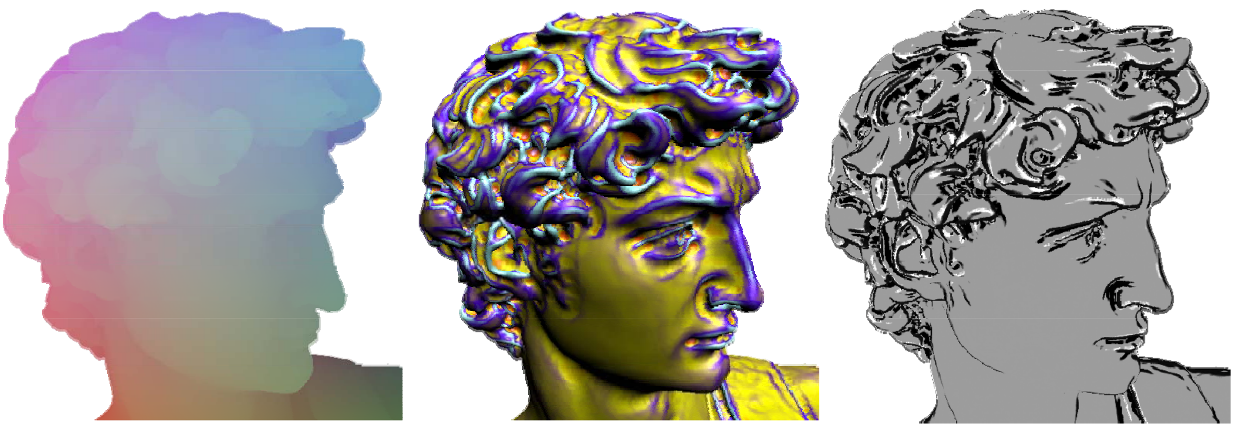
sampling rate 1/5 voxel (no adaptive sampling)

Intersection Refinement (3)

Sampling distance 1.0, 24 fps   Sampling distance 5.0, 66 fps

Deferred Isosurface Shading

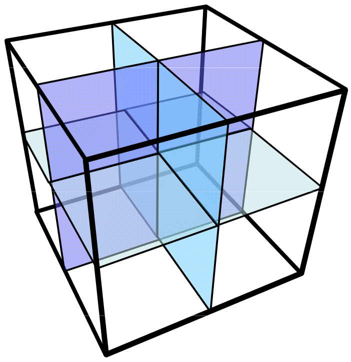
- Shading is expensive
- Full ray casting step computes only intersection image



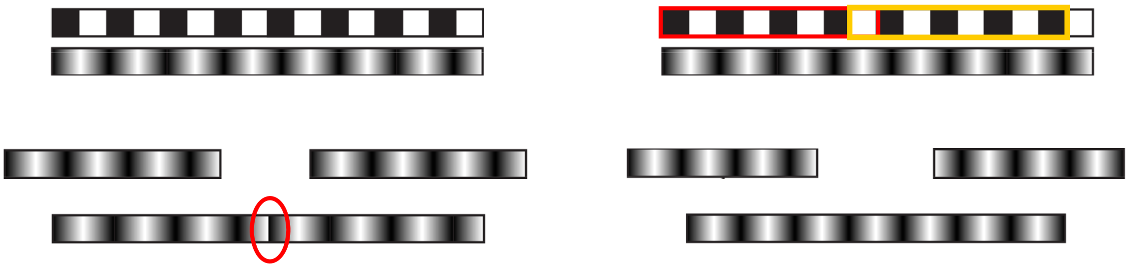
Memory Management

- What happens if data set is too large to fit into local GPU memory?

➡ Divide data set into smaller chunks (bricks)



One plane of voxels must be duplicated for correct interpolation across brick boundaries



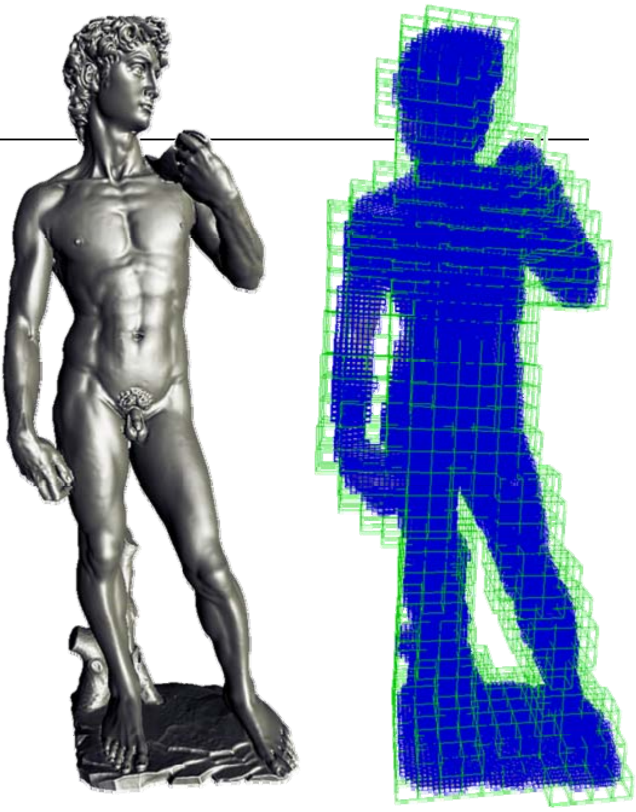
incorrect interpolation!

Bricking

- **Combine bricks for memory management**

with

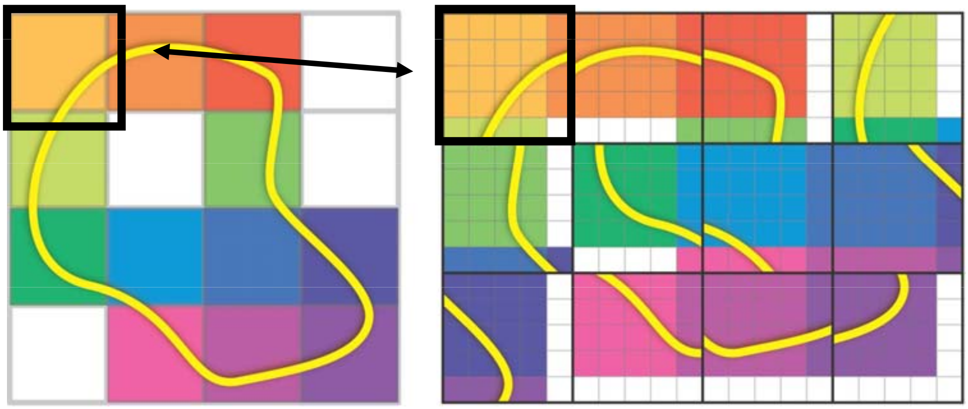
- **Smaller blocks for object-order empty space skipping**



1536x576x352

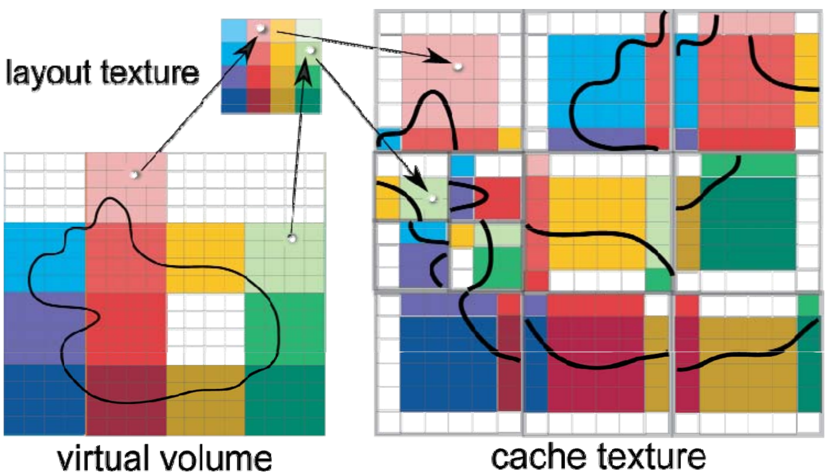
Bricked Single-Pass Casting (1)

- Duplicate neighbor voxels for filtering
- Store n^3 bricks as $(n+1)^3$
  - 10% overhead with 32^3 bricks
- Pack needed bricks into single 3D texture

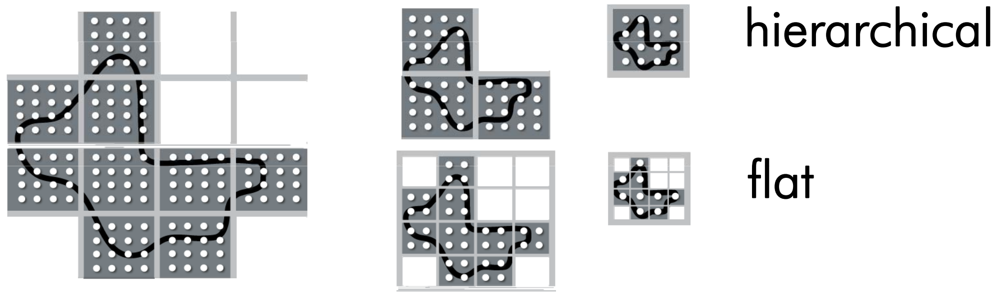


Bricked Single-Pass Casting (2)

- Layout/index texture for addr. translation
- Supports multi-resolution rendering
- Map virtual volume coords to physical tex

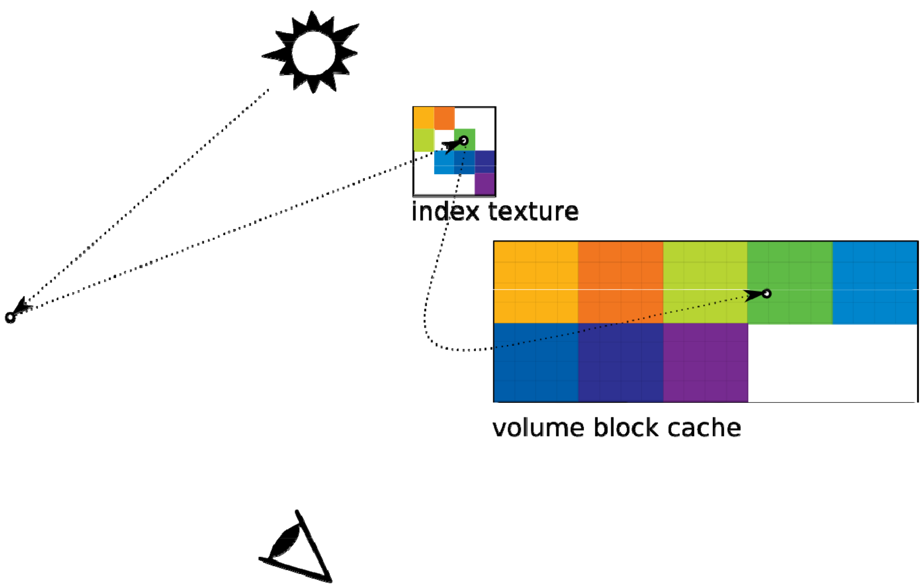


Flat vs. Hierarchical Bricking

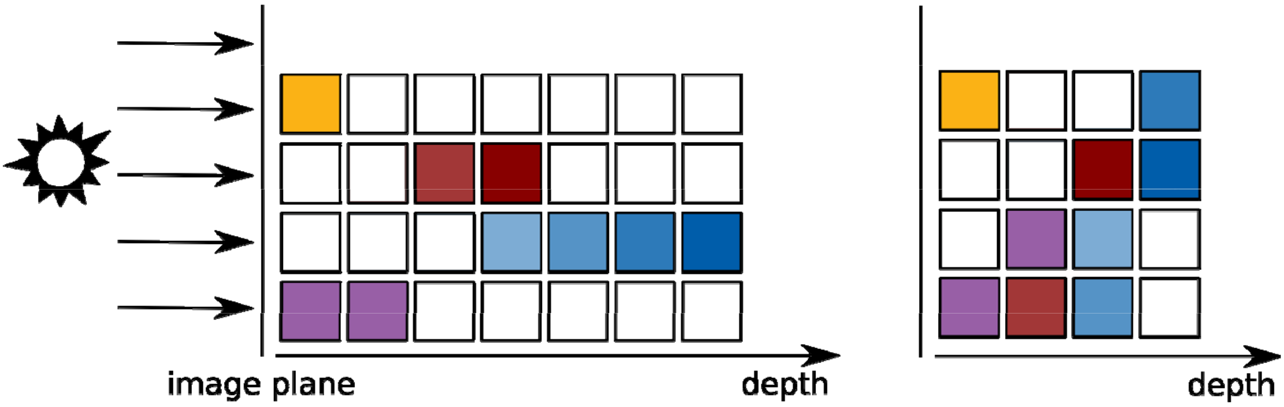


	flat	hierarchical
Number of bricks	⇔	⇩
Texture size of brick	⇩	⇔
Physical extent of brick	⇔	⇧

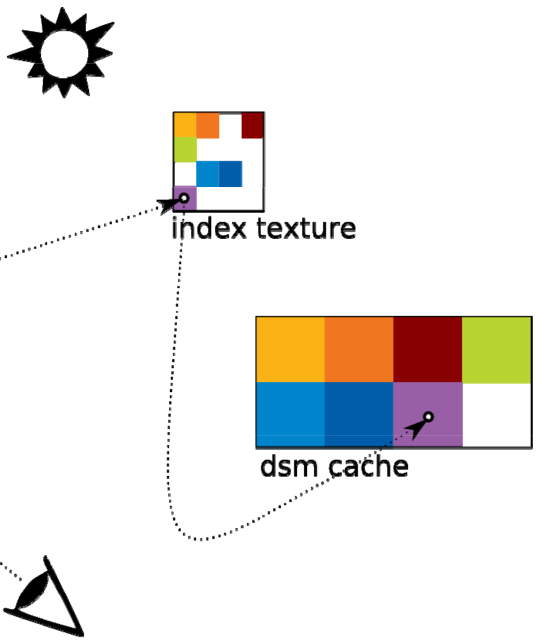
Shadow Memory Managm. (1)



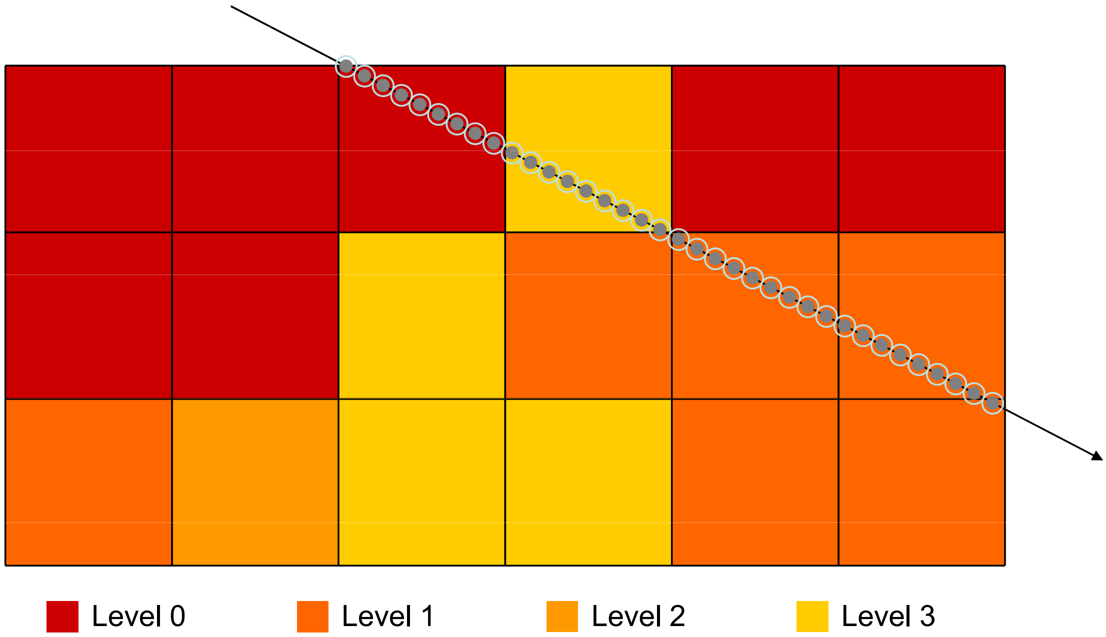
Shadow Memory Managm. (2)



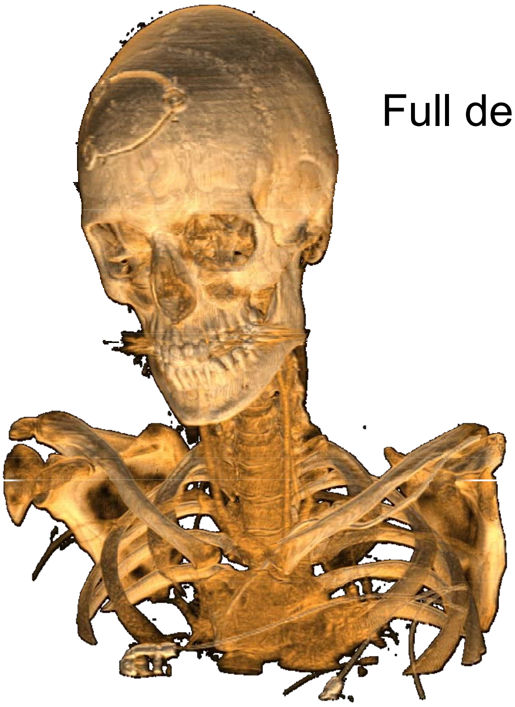
Shadow Memory Managm. (3)

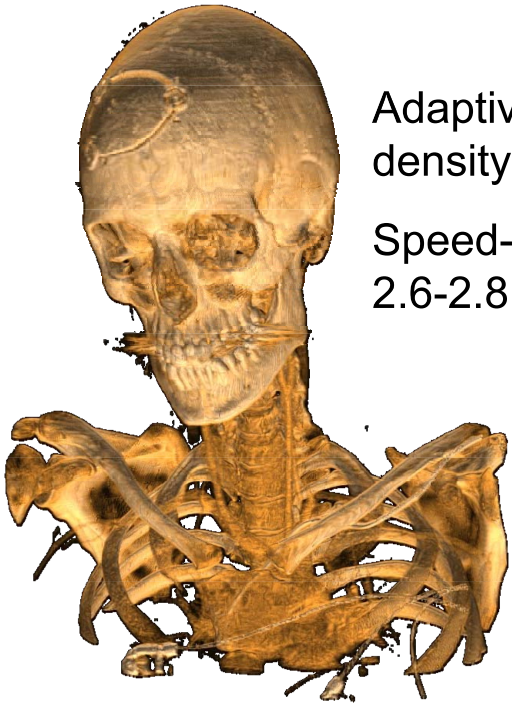


Adaptive Volume Sampling



Adaptive Volume Sampling

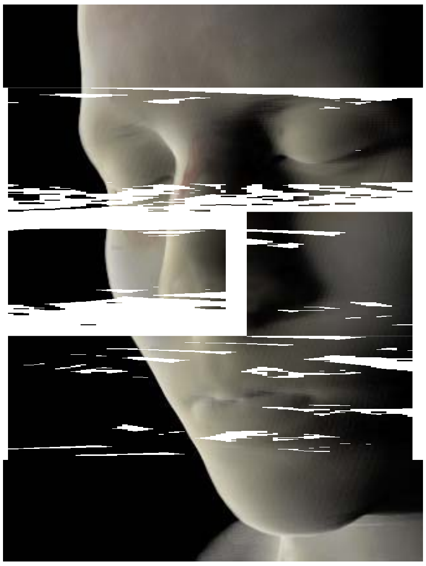


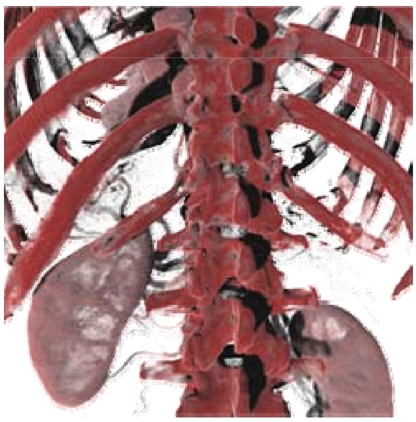


Conclusions

- Ray casting has become the most important GPU volume rendering technique
  - Very flexible and easy to implement
  - Now with advanced lighting in real time
- Mixing image-order and object-order approaches is well suited to GPUs
- Flexible memory management for both rendering and lighting

Thank You!





Acknowledgments
- Christof Rezk-Salama, Patric Ljung, Henning Scharsach, Daniel Weiskopf

ADVANCED ILLUMINATION TECHNIQUES FOR GPU-BASED VOLUME RAYCASTING
Eurographics 2009